ECS 30 **Homework #7** **Spring 2011**

Due: - gdb tutorial script file due electronically Friday, May 20th, at 11:50 pm;
- Written Exercises due Monday, May 23rd, at 4 pm in Homework box in 2131 Kemper;
- Program due electronically Monday, May 23rd, at 11:50 pm.

gdb Tutorial (15 pts): Go through Sean Davis's gdb tutorial, linked on the class web page. Use the script command in UNIX to document your work as follows. Just before you start the tutorial at the prompt issue the command:

```
script gdbtut.txt
```

This command will record everything that is typed on your screen in the file unixtut.txt until you issue the command exit. Submit the scriptfile gdbtut.txt to cs30, directory hw7 using handin, as follows: **`handin cs30 hw7 gdbtut.txt`**

Written Exercises (30 pts): The written exercises should be typed and each page should have at the top your name and ID#, section, and hw#. Handwritten answers will not be graded.

J&K, 10.1.2, 10.1.12, 10.1.20, 10.2.2, 10.3.4, 10.3.10, 10.4.4, 10.4.6, 10.5.4, 10.5.8, 10.5.10, 10.6.4, 10.9.2.

Program (55 pts): Handin a makefile and source code files, and have your makefile produce the executable file **`dating`** (use **`all`** … at the top). The third line in the source code files must contain the author of the file, ID, and section #. Use the *handin* program for electronic submission, described in the UNIX tutorial. For this homework use:

```
handin cs30 hw7 Makefile <your file1> <your file2> …
```

The date and time your files are created in the cs30 directory will be counted as your submit times. If those times are later than 11:50 pm on the due date your submissions will be considered late.

Dating Service (adapted from Exercise 10.10 in J&K, page 547)

The Happy-Ever-After Dating Service maintains records of individuals looking for dates. Each record tracks an individual's sex (M or F), social security number (SSN, 9 digits), name (< 20 letters), age (>18 and <85), occupation (<20 letters), yearly income (<100,000,000), major and minor hobbies (<20 letters), height (in inches), weight (in lbs), and religion (<20 letters). The records are sorted by SSN. The service matches individuals as follows:

Two individuals match if they are opposite in gender and either
(a) are within 10 years of one another's age and within $10,000 of each other's income, or
(b) share both a major and a minor hobby, or
(c) are within 10% of one another's weight and height, and share religion.

Write a program that asks the user for a social security number and the name of a file which contains individuals' records (< 1000 records). Each record in the file contains the above 11 attributes, each in a separate line; there is an empty line between consecutive records. The program should then read in all the records into an array of structures and find all matching individuals to the one with the specified SSN. Then, the names of the matching individuals, if any, should be printed out in ascending order of their SSN's.

Make sure the output of your program matches exactly the output below of my executable located at **`/home/cs30/public/hw7/dating`** on the csif machines. In the same directory you will find a file **`clients.txt`** which contains example records.

```
[cs30@pc50 hw7]$ ./dating
SSN: 111223333
Name of file of records: clients.txt

Matching clients:

Ann Robinson
Fran Gehry

[cs30@pc50 hw7]$ ./dating
SSN: 123456789
Name of file of records: clients.txt

Matching clients:

[cs30@pc50 hw7]$ ./dating
SSN: 111111111
Name of file of records: clients.txt

SSN 111111111 not found in file “clients.txt”.
```